Jet Propulsion Laboratory
California Institute of Technology

NASA Analogy Software Web-Based Cost Analysis Tool

Advances in Cost Analysis Methods

AIAA Economics Workshop, March 29th 2017, El Segundo, CA

Michael Saing

Dr. Jairus Hihn

Elinor Huntington

Alex Lumnah

Jet Propulsion Laboratory, California Institute of Technology

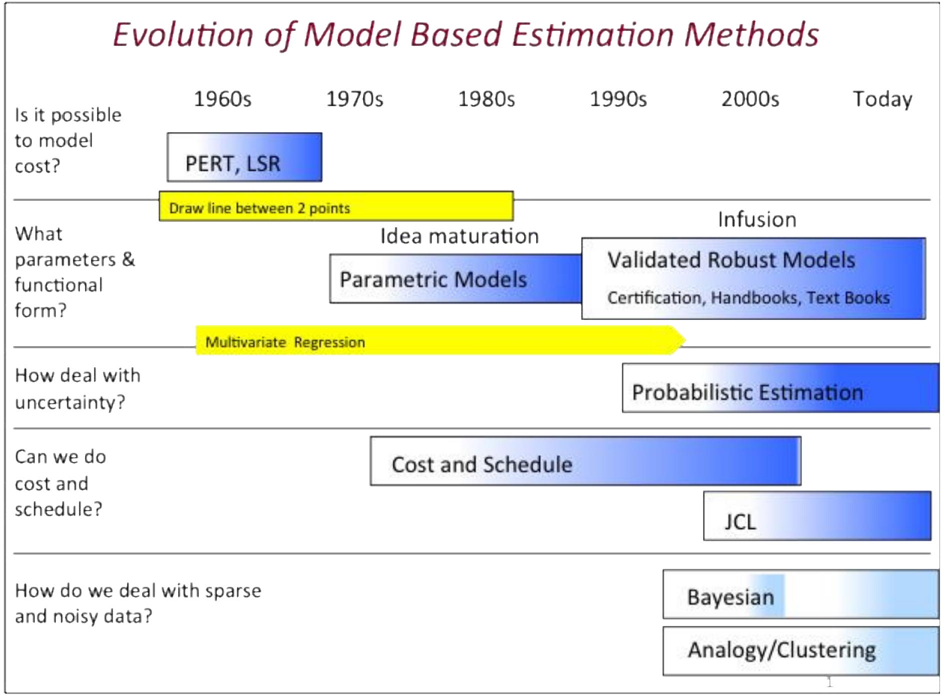
James Johnson

National Aeronautics and Space Administration

Cost Estimating

- *“How much will it cost?”*
- Why is it important
  - ”Am I in the right ball park range?” Get closer to accurate estimate
  - Keep one from making a big mistake, such as under estimating
    - NASA’s estimated cost overrun

Formal Analogy and Bayesian Models are a Natural Next Step in the Evolution Cost Modeling and Analysis



Why explore alternative modeling methods?

- For most of our history the cost community has relied upon regression type modeling methods
  - Regression method have the underlying assumption of
    - clean and complete data with large sample sizes
  - Cost data suffers from sparseness, noise, and small sample sizes
  - There are alternative methods that handle these conditions better then regression
- New cost method is built around a spectral clustering algorithm that can be used to estimate software size and effort that is effective for
  - small sample sizes
  - noisy data
  - and uses high level systems information (Symbolic Data)

Data – Missions

- Over 60 data total
- Cluster Analysis:
  - 34 missions
- Regression Analysis:
  - 37 Missions
- Data:
  - NASA 93 - Historical NASA data originally collected for ISS (1985-1990) and extended for NASA IV&V (2004-2007)
  - NASA software inventory
  - Jairus 30+years in SW data collection

Data Item	Number of Missions (Current - 2017)	Number of Missions (2016)
Total development effort in work months	36	28
Flight Software Development Cost	37	30
Flight System Development Cost	37	30
Logical Lines of code (LOC)		
Delivered LOC	49	36
Inherited LOC (Reused plus Modified reused lines)	43	36
COCOMO model inputs (See Appendix A for the parameter definitions) - Translated from CADRe which has SEER model inputs because the SEER data items are very sparse in CADRe	19	19
System parameters * (See Appendix B parameter definitions)		
Mission Type (deep-space, earth-moon, rover-lander, observatory)	49	39
Multiple element (probe, etc.)	49	39
Number of Instruments	49	39
Number of Deployables	49	39
Flight Computer Redundancy (Dual Warm, Dual Cold, Single String)	49	39
Software Reuse (Low, Medium, High)	41	36
Software Size (Small, Medium, Large, Very Large)	41	36

Mission	ASCoT	Regression
Cassini	X	
Contour		X
Dawn	X	X
Deep Impact	X	X
DS1	X	X
Genesis	X	X
GLL	X	
JUNO	X	X
LADEE		X
MAP		X
Mars Odyssey	X	
Maven	X	
Messenger	X	X
MRO	X	X
NEAR	X	X
New Horizons	X	X
OSIRIS REX	X	X
Stardust	X	
Van Allen Probe (RBSP)	X	X
GRO	X	
HST	X	
Kepler	X	
Stereo	X	X
WISE		X
AIM		
Aqua		
EO1		
FAST		X
GALEX		
GEMS	X	
GEOTAIL		
GLAST		
GLORY	X	X
GOES R	X	
GPM Core	X	X
Grail	X	X
IBEX		X
IRIS		
LCROSS		
LDCM		
LRO	X	X
NOAA-N-Prime		
NPP		
NuStar	X	X
OCO	X	X
OCO 2		X
OCO 3		
RHESSI		
SAMPEX		X
SDO	X	X
SMAP	X	
SWAS		X
TIMED	X	X
TRACE		X
TRMM		X
WIRE		X
MER	X	X
MPF	X	X
MSL	X	X
Insight	X	X
Phoenix	X	X

System Descriptor Details (Example)

- Systems Level Descriptors used as symbolic data input
  - Category derived from the count/cost #'s (i.e. small, medium, large, etc…)

Software Delivered Code		
Values	**Description**	**Example**
Small	Delivered logical lines of code is < 50 KSLOC	Small earth orbiters
Medium	Delivered logical lines of code is < 50 KSLOC and < 120 KSLOC	LRO, Kepler
Large	Delivered logical lines of code is < 120 KSLOC and < 220 KSLOC	LCROSS, SMAP, Phoenix
Very Large	Delivered logical lines of code is > 220 KSLOC	Rovers

Inheritance		
Values	**Description**	**Example**
Low to None	Total Inherited code, including modified code is < 10% of delivered code.	MER, TIMED, LRO
Low	Total Inherited code, including modified code is between 10% to 20% of delivered code.	Deep Impact, New Horizons
Medium	Total Inherited code, including modified code is >= 20% and < 50% of delivered code.	Messenger, MRO
High	Total Inherited code, including modified code is >= 50% and < 80% of delivered code.	JUNO, SDO, GPM core
Very High	Total Inherited code, including modified code is a minimum of 80% of delivered code.	MAVEN, Grail, NOAA-N-Prime

Total Mission cost		
Values	**Description**	**Example**
Small	Total Mission cost including operations in FY15 dollars is > $120M and < $220 million	Wise, small earth orbiters
Medium	Total Mission cost including operations in FY15 dollars is > $220 million and < $600 million	Discovery class missions
Large	Total Mission cost including operations in FY15 dollars is > $600 million and < $1.1 billion	New Frontiers class missions
Very Large	Total Mission cost including operations in FY15 dollars is > $1.1 billion	Large assigned mission, MSL

Data – Mission Descriptors

Categorical

Mission Type	Software Size					
	#Rec.	Very Low to None	Low	Med	High	Very High
Earth/Lunar Orbiter	22	3	13	6	0	Medium
Observatory	6	1	5	0	0	Medium
Deep Space	16	2	4	7	3	Large
In Situ	5	0	1	2	2	Large

Mission Type	Inheritance						
	#Rec.	Very Low to None	Low	Med	High	Very High	Med.
Earth/Lunar Orbiter	18	4	0	4	4	6	High
Observatory	5	0	1	2	1	1	Low
Deep Space	15	2	3	2	3	5	High
In Situ	5	2	1	0	1	1	Very Low/ None

Mission Type	Flight Computer Redundancy				
	#Rec.	Single String	Dual-String Cold	Dual-String Warm	Median
Earth/Lunar Orbiter	22	14	8	0	Single String
Observatory	6	1	5	0	Dual String Cold
Deep Space	16	1	13	2	Dual String Cold
In Situ	5	1	0	4	Dual String Warm

Numerical

Mission Type	EFFORT (months)				
	# Records	Median	S.D.	Avg.	Range
Earth/Lunar Orbiter	22	584	354	651	100 – 1,190
Observatory	5	492	631	742	233 – 1,830
Deep Space	17	637	375	686	48 – 1,436
In Situ	5	1,080	555	1,232	634 – 1,888

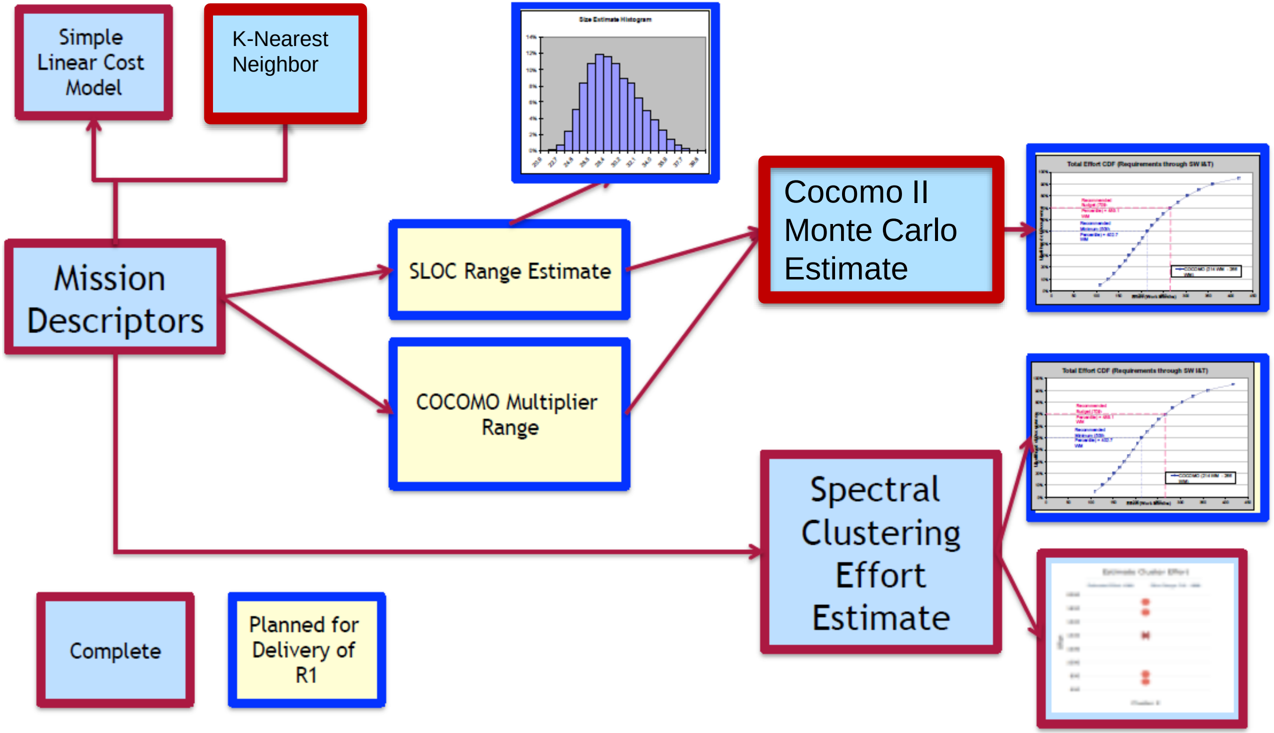
Mission Type	Logical Delivered LOC				
	#Rec.	Median	S.D.	Avg.	Range
Earth/Lunar Orbiter	22	96,000	41,432	101,821	12,000 – 170,000
Observatory	5	107,000	95,548	23,000	23,000 – 280,000
Deep Space	17	122,000	75,431	24,000	24,000 – 289,900
In Situ	5	205,000	145,334	94,300	94,300 – 475,000

Mission Type	Productivity (Logical Del/month)				
	# Records	Median	S.D.	Avg.	Range
Earth/Lunar Orbiter	22	191	214	260	65 – 823
Observatory	5	244	192	238	46 – 460
Deep Space	17	208	168	262	37 – 615
In Situ	5	249	81	212	87 - 292

Introduction to NASA ASCoT

- NASA Analogy Software Costing Tool (ASCoT)
- The purpose of the model is to
  - Supplement current estimation capabilities
  - *Be effective in the very early lifecycle when our knowledge is fuzzy*
    - uses high level systems information (Symbolic Data)
  - *Be usable by Cost Estimators, Software Engineers and Systems Engineers*
- The *NASA Software CER Development Task is funded by the NASA HQ Strategic Investment Division to develop a software cost model that*
  - *Can be used in the early lifecycle*
  - *Can be used effectively by non-software specialists*
  - *Uses data from NASA in-house built and funded software "projects"*
    - *Supplement to current modeling and bottom up methods not a replacement*
  - *Acceptable for use with both the cost and software communities*

Model Architecture



Key Analysis Components

Cluster Analysis	Regression Analysis	COCOMOII Analysis	KNN
-Spectral Clustering -Development Effort Estimate	-Linear Regression -Development Estimate	-Verified Reproduction -SLOC/ Cost/Effort	-Nearest Neighbor -Development Cost/Effort

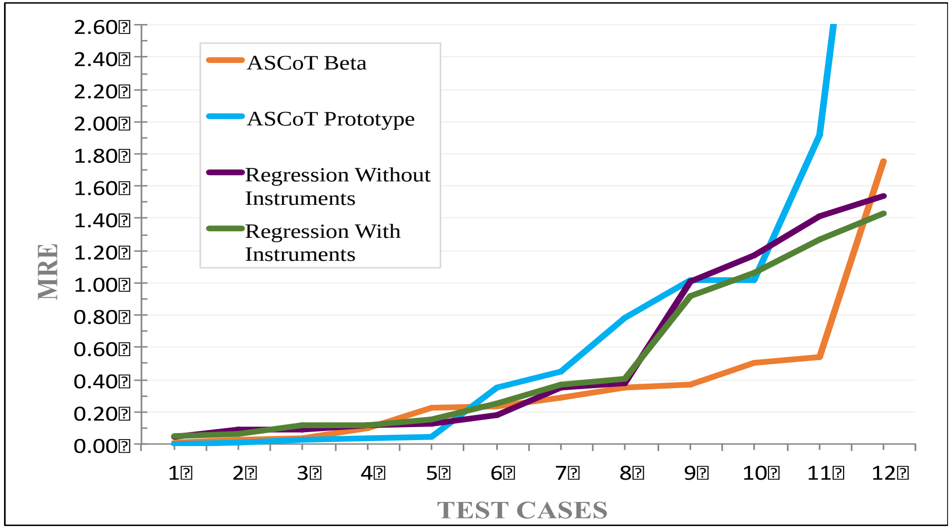
- Cluster & Regression Analysis components listed rely on high level Mission Descriptors such as # of Instruments and Mission Type
    - Ref. System Parameters with Definitions and Examples
- COCOMOII is a reproduction and uses traditional inputs
- KNN predicts the numerical target based on a similarity measure (e.g. distance)

Model Performance Comparison

- Magnitude of Relative Error (MRE) as a metric for evaluating model performance:
  MRE = (Predicted – Actual) / Actual
- MRE and Pure clustering
  - Median distance between two clusters is best
    - Produces lower over all MRE. Median MRE is not sensitive to outliers, and therefore is more appropriate as a measure of the central tendency of a skewed distribution
  - Median measures always win
    - Has implications for our commonly used regression based models which are regression to the mean
- MRE used to measure against the performance of:
  - Simple Linear Regression Model
  - Spectral Clustering Model

MRE Results

- ASCoT Beta performs best. The smaller % percentage error does best.
- Cluster for each test cases is 3-4 data per cluster

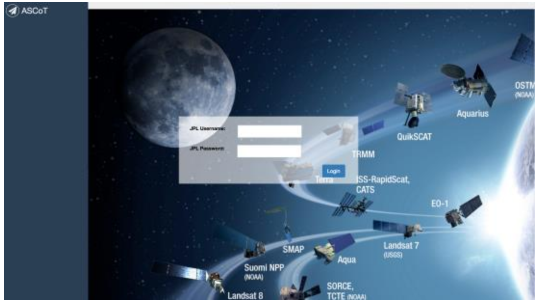


MRE by Rank Order and Model Version

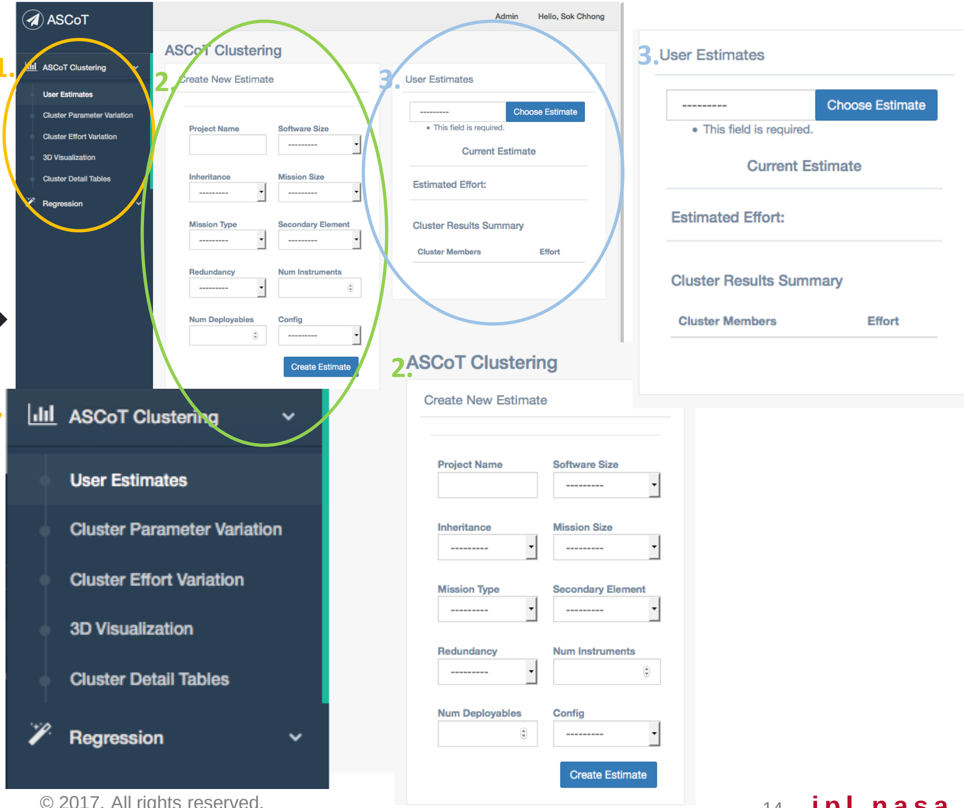
	Regression Without Instruments	Regression With Instruments
	4%	5%
	9%	6%
	9%	11%
	11%	12%
	13%	15%
	18%	26%
	35%	37%
	38%	40%
	101%	92%
	117%	106%
	141%	127%
	154%	143%
Median MRE	27%	31%
Mean MRE	54%	52%

ASCoT DEMO

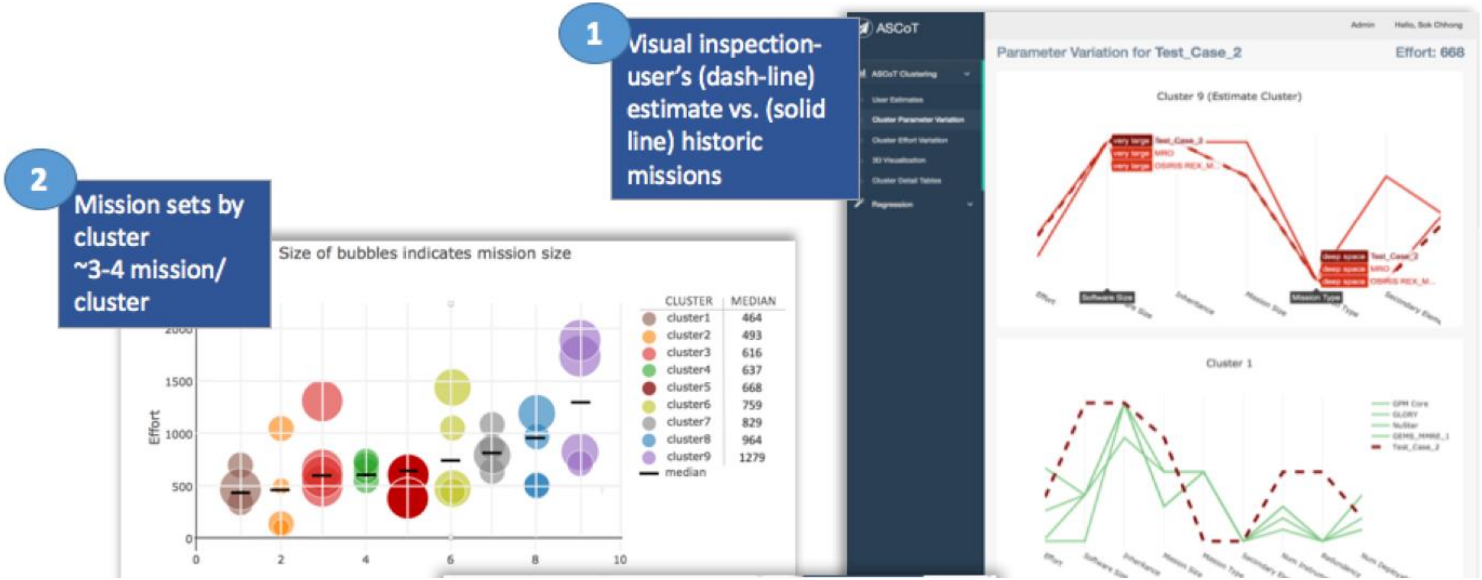
Tool’s User Interface



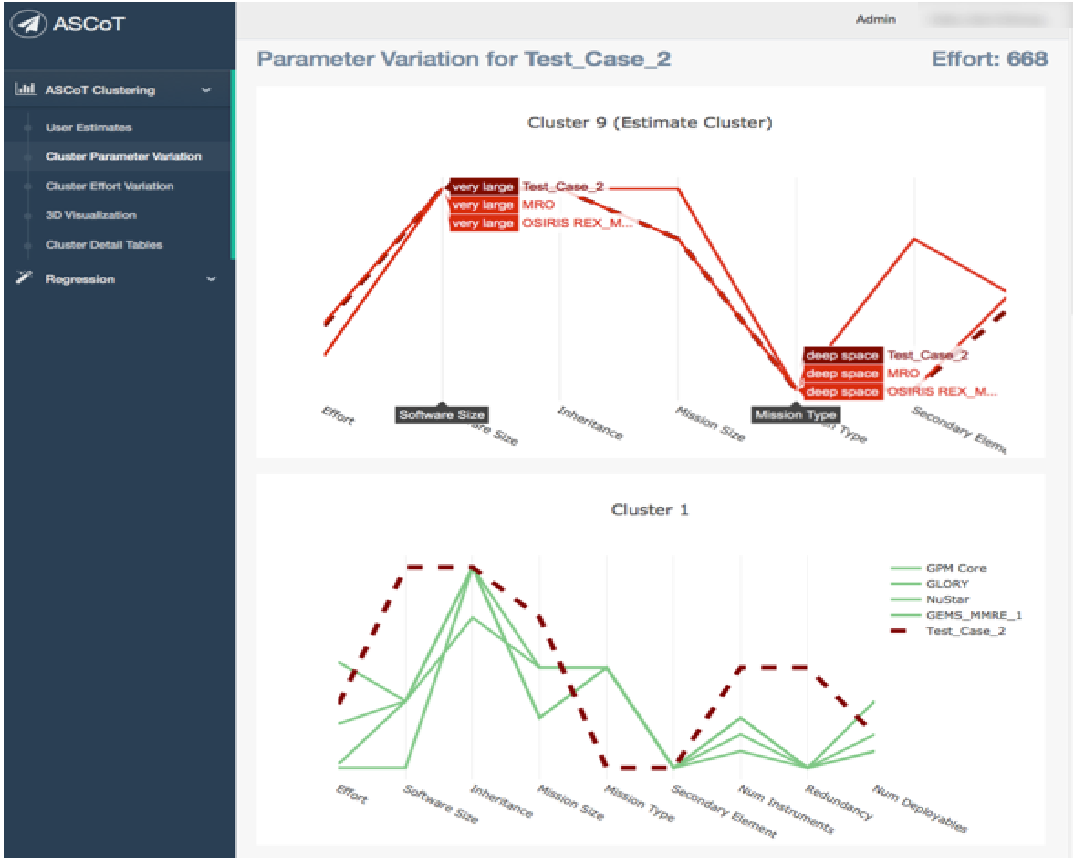
Web
Log-in



Cluster Estimating Results



ASCoT Variation by Mission Descriptors



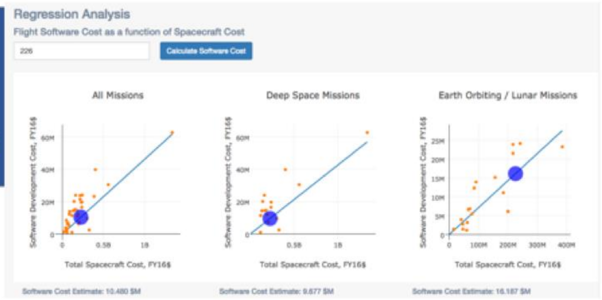
Mission Descriptors

- Effort
- Software Size
- Inheritance
- Mission Size
- Secondary Element
- Number of Instruments
- Redundancy
- Number of Deployments

Regression Tool in ASCoT

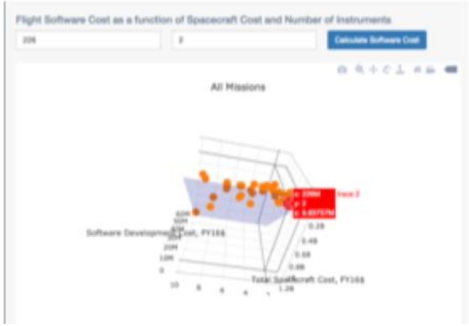
1

2D Plots by Mission Type based on S/C Cost
x=S/C dev. Cost
Y=SW Cost



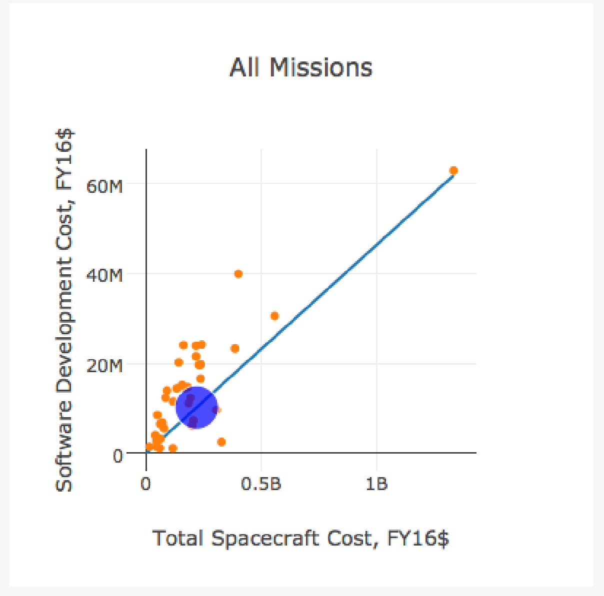
2

3D rotational plot
x= S/C dev. Cost
y=# instruments
Z=SW dev. Cost



Regression Analysis

- User input spacecraft cost to get estimated software development costs
- Linear regression trend line
- Phase B-D Costs



More References and Readings

- If you're interested in reading more about this and want to see more detailed methodology, performance and model, please contact Dr. Jairus Hihn at Jairus.m.hihn@jpl.nasa.gov

Questions

- Acknowledgements
  - Dr. Jairus Hihn, JPL, Group Supervisor
  - James Johnson, NASA HQ, Strategic Investment Division
  - Elinor Huntington, JPL
  - Alex Lumnah, JPL

Author’s Note

- Proper credit of third party cited references/materials has been cited and obtained with permission

Back-up

Data Mining Methods

- Data mining techniques provided us with the rigorous tool set
- we needed to explore the many dimension of the problem we
- were addressing in a repeatable manner
  - Analyze standard and non-standard models
    - Is there a best functional form
- Perform exhaustive searches over all parameters and
- records in order to guide data pruning
  - Rows (Stratification)
  - Columns (variable reduction)
- Measure model performance by multiple measures
  - R^2, MRE, Pred, F-test, etc.
- Is there a 'best' way to tune or calibrate a model

Spectral Clustering

- PCA finds eigenvectors in numerical data
- Spectral Clustering
  - Spectral Clustering is like PCA on steroids but uses an eigenvector approximation method
  - Recursively splits the data on synthesized dimension of greatest variance/spread
- Why use it
  - Can handle numerical and symbolic data
  - Can work on small, sparse and somewhat noisy data sets but also works well on large consistent data sets
  - Can use as estimator with partial information

Effort Estimation with Data Mining Methods References

"Active Learning and Effort Estimation: Finding the Essential Content of Software Effort Estimation Data" by Ekrem Kocaguneli and Tim~Menzies and Jacky Keung and David Cok and Ray Madachy. IEEE Transactions on Software Engineering (pre-print) 2013 . .

"Finding conclusion stability for selecting the best effort predictor in software effort estimation" by J. Keung and E. Kocaguneli and T. Menzies. Automated Software Engineering pages 1-25 May 2012 . Available from http://menzies.us/pdf/12findstable.pdf .

"Exploiting the Essential Assumptions of Analogy-Based Effort Estimation" by E. Kocaguneli and T. Menzies and A. Bener and J. Keung. IEEE Transactions on Software Engineering pages 425-438 2012 . Available from http://menzies.us/pdf/11teak.pdf .

"Local vs. Global Lessons for Defect Prediction and Effort Estimation" by Menzies, T. and Butcher, A. and Cok, D. and Marcus, A. and Layman, L. and Shull, F. and Turhan, B. and Zimmermann, T.. IEEE Transactions on Software Engineering pages 1 2012 . Available from http://menzies.us/pdf/12localb.pdf .

"Kernel methods for software effort estimation" by E. Kocaguneli and T. Menzies and J. Keung. Empirical Software Engineering pages 1-24 2011

"On the Value of Ensemble Effort Estimation" by Kocaguneli, E. and Menzies, T. and Keung, J.. IEEE Transactions on Software Engineering 2011

"Exploring the Effort of General Software Project Activities with Data Mining" by Topi Haapio and Tim Menzies. International Journal of Software Engineering and Knowledge Engineering pages 725-753

"Stable Rankings for Different Effort Models" by Tim Menzies and Omid Jalali and Jairus Hihn and Dan Baker and Karen Lum. Automated Software Engineering December 2010 . Available from http://menzies.us/pdf/10stable.pdf .

"Case-Based Reasoning for Reducing Software Development Effort" by Adam Brady and Tim Menzies and Oussama El-Rawas and Ekrem Kocaguneli and Jacky Keung. Journal of Software Engineering and Applications 2010 . Available from http://menzies.us/pdf/10w0.pdf .

"A Second Look at Faster, Better, Cheaper" by Oussama El-Rawas and Tim Menzies. Innovations Systems and Software Engineering pages 319-335 2010 . Available from http://menzies.us/pdf/10bfc.pdf .

"Explanation vs Performance in Data Mining: A Case Study with Predicting Runaway Projects" by Tim Menzies and O. Mizuno and Y. Takagi and Y. Kikuno. Journal of Software Engineering and Applications pages 221-236 November 2009

"Accurate Estimates Without Local Data?" by Tim Menzies and S. Williams and Oussama El-Rawas and D. Baker and B. Boehm and J. Hihn and K. Lum and R. Madachy. Software Process Improvement and Practice pages 213-225 July 2009 . Available from http://menzies.us/pdf/09nodata.pdf .

"Selecting Best Practices for Effort Estimation" by Menzies, Tim and Chen, Zhihao and Hihn, Jairus and Lum, Karen. IEEE Transactions on Software Engineering pages 883--895 doi = 10.1109/TSE.2006.114 issue = 11 2006